

# Simulation de Modèles Discrets

Fabien Givors

*d'après les cours d'Alberto Dennunzio*

Département d'Informatique  
Université de Nice-Sophia Antipolis

- 1 Cliquer sur “Button” dans la barre d’outils

- 1 Cliquer sur “Button” dans la barre d’outils
- 2 Cliquer dans la partie blanche de l’interface pour le faire apparaître

# Créer un bouton

- 1 Cliquer sur “Button” dans la barre d’outils
- 2 Cliquer dans la partie blanche de l’interface pour le faire apparaître
- 3 Éditer la boîte du bouton : nom, commandes, ...

# Exemple : Setup

- ① Édition de la boîte du bouton :
  - Display name : setup
  - Commands : *setup*
  - Agent(s) : Observer

# Exemple : Setup

## ① Édition de la boîte du bouton :

- Display name : setup
- Commands : *setup*
- Agent(s) : Observer

**Remarque** : il faut définir la commande setup !

## Exemple : Setup

① Édition de la boîte du bouton :

- Display name : setup
- Commands : *setup*
- Agent(s) : Observer

**Remarque** : il faut définir la commande setup !

② Aller dans l'onglet code et saisir :

```
to setup
  clear-all
  create-turtles 100
  ask turtles [ setxy random-xcor random-ycor ]
end
```

## Exemple : Setup

① Édition de la boîte du bouton :

- Display name : setup
- Commands : *setup*
- Agent(s) : Observer

**Remarque** : il faut définir la commande setup !

② Aller dans l'onglet code et saisir :

```
to setup
  clear-all
  create-turtles 100
  ask turtles [ setxy random-xcor random-ycor ]
end
```

**On a créé le contrôle setup utilisant la procédure *setup* !**

```
to    nom_de_la_procédure [ input1 ...]  
      ...  
      commandes  
      ...  
end
```

Les commandes peuvent être pré-définies ou définies par d'autres procédures.

La procédure peut (ou non) avoir des inputs

# Exemple de procédure : Setup

```
to setup  
  clear-all  
  create-turtles 100  
  ask turtles [ setxy random-xcor random-ycor ]  
end
```

# Exemple de procédure : Setup

```
to setup  
  clear-all  
  create-turtles 100  
  ask turtles [ setxy random-ycor random-xcor ]  
end
```

*clear-all* : le monde est effacé

## Exemple de procédure : Setup

```
to setup  
  clear-all  
  create-turtles 100  
  ask turtles [ setxy random-xcor random-ycor ]  
end
```

*clear-all* : le monde est effacé

*create-turtles 100* : 100 turtles sont créés dans (0,0)

## Exemple de procédure : Setup

```
to setup  
  clear-all  
  create-turtles 100  
  ask turtles [ setxy random-xcor random-ycor ]  
end
```

*clear-all* : le monde est effacé

*create-turtles 100* : 100 turtles sont créés dans (0,0)

*setxy random-xcor random-ycor* : chaque turtle récupère ses coordonnées en utilisant les rapporteurs *random-xcor* et *random-ycor*

## Exemple de procédure : Setup

```
to setup  
  clear-all  
  create-turtles 100  
  ask turtles [ setxy random-pxcor random-ycor ]  
end
```

*clear-all* : le monde est effacé

*create-turtles 100* : 100 turtles sont créés dans (0,0)

*setxy random-pxcor random-ycor* : chaque turtle récupère ses coordonnées en utilisant les rapporteurs *random-pxcor* et *random-ycor*

*random-pxcor* : rapporte un nombre entre *min-pxcor-0.5* et *max-pxcor+0.5*

fonction qui rend une valeur

fonction qui rend une valeur

- ▶ peut (ou pas) avoir des input
- ▶ peut être prédéfinie ou définie par le programmeur

fonction qui rend une valeur

- ▶ peut (ou pas) avoir des input
- ▶ peut être prédéfinie ou définie par le programmeur

```
to-report nom_du_reporteur [ input1 ...]  
    ...  
    report valeur_à_reporter  
end
```

fonction qui rend une valeur

- ▶ peut (ou pas) avoir des input
- ▶ peut être prédéfinie ou définie par le programmeur

```
to-report nom_du_reporteur [ input1 ...]  
    ...  
    report valeur_à_reporter  
end
```

*random-xcor* : prédéfini sans input

*random* : prédéfini avec input (instance d'appel : *random 360*)

# Exemple de procédure : Go

- ① Édition de la boîte du bouton :
  - Display name : go
  - Commands : *go*
  - Agent(s) : Observer

## Exemple de procédure : Go

- ① Édition de la boîte du bouton :
  - Display name : go
  - Commands : *go*
  - Agent(s) : Observer

**Remarque** : il faut définir la commande go !

# Exemple de procédure : Go

## ① Édition de la boîte du bouton :

- Display name : go
- Commands : *go*
- Agent(s) : Observer

**Remarque** : il faut définir la commande go !

## ② Dans l'onglet Code, entrer :

```
to go
  move-turtles
end

to move-turtles
  ask turtles [
    right random 360
    forward 1
  ]
end
```

## Exemple : Go

**On a créé le contrôle go utilisant la procédure go !**

La procédure *go* utilise la procédure *move-turtles*.

```
to go
  move-turtles
end

to move-turtles
  ask turtles [
    right random 360
    forward 1
  ]
end
```

## Exemple : Go

**On a créé le contrôle go utilisant la procédure go !**

La procédure *go* utilise la procédure *move-turtles*.

```
to go
  move-turtles
end

to move-turtles
  ask turtles [
    right random 360
    forward 1
  ]
end
```

*right random 360* : chaque turtle tourne à droite du nombre de degrés reportés par le reporteur *random* (appel à *random 100* )

*forward 1* : chaque turtle avance de 1.

# On enrichit Setup

```
to setup
  clear-all
  setup-patches
  setup-turtles
end

to setup-patches
  ask patches [ set pcolor green ]
end

to setup-turtles
  create-turtles 100
  ask turtles [ setxy random-xcor random-ycor ]
end
```

- ▶ globales

*global* [nom\_de\_variable]

# Créer des variables

- ▶ globales

*global* [nom\_de\_variable]

- ▶ propriétaires (paramètres des agents)

*turtles-own* [nom\_de\_variable]

*patches-own* [nom\_de\_variable]

*link-own* [nom\_de\_variable]

# Créer des variables

- ▶ globales

*global* [nom\_de\_variable]

- ▶ propriétaires (paramètres des agents)

*turtles-own* [nom\_de\_variable]

*patches-own* [nom\_de\_variable]

*link-own* [nom\_de\_variable]

définir/changer une valeur d'une variable :

*set* nom\_de\_variable valeur

## Exemple

```
turtles-own [energy]

to go
  move-turtles
  eat-grass
end

to eat-grass
  ask turtles [
    if pcolor = green [
      set pcolor black
      set energy (energy + 10)
    ]
  ]
end
```

*if* condition [commandes]

*ifndef* condition [commandes1] [commandes2]

## On enrichit le modèle

```
to move-turtles
  ask turtles [
    right random 360
    forward 1
    set energy energy - 1
  ]
end
```

- 1 Choisir et cliquer sur “Monitor” dans la barre des outils

- ① Choisir et cliquer sur “Monitor” dans la barre des outils
- ② Cliquer dans la partie blanche de l’interface pour le faire apparaître

- 1 Choisir et cliquer sur “Monitor” dans la barre des outils
- 2 Cliquer dans la partie blanche de l’interface pour le faire apparaître
- 3 Éditer la boîte du bouton : nom, rapporteur, ...

## Exemple : Monitor “nombre de turtles”

Edite la boîte du bouton :

- Display name : nombre de turtles
- Reporter : *count turtles*

*count* agentset :

rapporte le nombre d'agents dans l'agentset (*turtles*, *patches*, *links*, un ensemble d'agents créé par le programmeur)

## Exemple : Monitor “patches verts”

Dans la boîte de dialogue du bouton :

- Display name : patches verts
- Reporter : *count patches with [pcolor = green]*

agentset *with* [boolean reporter] :

rapporte un nouvel agentset qui contient seulement les agents parmi l'agentset donné qui rapportent vrai.

- 1 Choisi "Switch" dans la barre d'outils

# Créer un Switch

- ① Choisir "Switch" dans la barre d'outils
- ② Cliquer dans la partie blanche de l'interface pour le faire apparaître

# Créer un Switch

- 1 Choisir "Switch" dans la barre d'outils
- 2 Cliquer dans la partie blanche de l'interface pour le faire apparaître
- 3 Éditer la boîte du switch : variable globale

## Exemple : Switch “show-energy?”

- 1 Éditer la boîte du switch :
  - Global variable : *show-energy?*
- 2 Entrer dans Code et réécrire *eat-grass* :

```
to eat-grass
  ask turtles [
    if pcolor = green [
      set pcolor black
      set energy (energy + 10)
    ]
    ifelse show-energy?
      [ set label energy ]
      [ set label "" ]
  ]
end
```

- 1 Choisir “Slider” dans la barre d'outils

- 1 Choisir “Slider” dans la barre d’outils
- 2 Cliquer dans la partie blanche de l’interface pour le faire apparaître

- 1 Choisir “Slider” dans la barre d’outils
- 2 Cliquer dans la partie blanche de l’interface pour le faire apparaître
- 3 Éditer la boîte du slider : variable globale, valeur de default, minimum, ...

## Exemple : Slider associé à *number*

- ① Dans la boîte du slider :
  - Global variable : *number*
  - Minimum : 0
  - Increment : 1
  - Maximum : 200
  - Value : 100

## Exemple : Slider associé à *number*

- ① Dans la boîte du slider :
  - Global variable : *number*
  - Minimum : 0
  - Increment : 1
  - Maximum : 200
  - Value : 100
- ② Entrer dans Code et réécrire *setup-turtles* :

```
to setup-turtles
  create-turtles number
  ask turtles [ setxy random-xcor random-ycor ]
end
```

# On enrichit le modèle

```
to go  
  move-turtles  
  eat-grass  
  reproduce  
  check-death  
  regrow-grass  
end
```

```
to reproduce
  ask turtles [
    if energy > 50 [
      set energy energy - 50
      hatch 1 [ set energy 50]
    ]
  ]
end

to check-death
  ask turtles [
    if energy <= 0 [ die ]
  ]
end

to regrow-grass
  ask patches [
    if random 100 < 3 [ set pcolor green ]
  ]
end
```

- 1 Choisir “Plot” dans la barre d’outils

# Créer un Plot

- 1 Choisir “Plot” dans la barre d’outils
- 2 Cliquer dans la partie blanche de l’interface pour le faire apparaître

# Créer un Plot

- 1 Choisir “Plot” dans la barre d’outils
- 2 Cliquer dans la partie blanche de l’interface pour le faire apparaître
- 3 Éditer la boîte du plot : nom, X axis label, ...

# Créer un Plot

- 1 Choisir “Plot” dans la barre d’outils
- 2 Cliquer dans la partie blanche de l’interface pour le faire apparaître
- 3 Éditer la boîte du plot : nom, X axis label, ...
- 4 Cliquer sur “Create” pour créer chacun des stylos désirés et en éditer le nom (dans la fenêtre apparue), la couleur, ...

# Créer un Plot

- 1 Choisir “Plot” dans la barre d’outils
- 2 Cliquer dans la partie blanche de l’interface pour le faire apparaître
- 3 Éditer la boîte du plot : nom, X axis label, ...
- 4 Cliquer sur “Create” pour créer chacun des stylos désirés et en éditer le nom (dans la fenêtre apparue), la couleur, ...
- 5 utilise les commandes
  - *set-current-plot* nom\_de\_plot
  - *set-current-plot-pen* nom\_de\_stylo
  - *plot* reporter

## Exemple : Plot “Totals”

Editer la boite du plot :

- Name : Totals
- X axis label : time
- Y axis label : totals

## Exemple : Plot “Totals”

Editer la boite du plot :

- Name : Totals
- X axis label : time
- Y axis label : totals

création de 2 stylos :

- Enter a name for the pen : turtles
- Color : black
- Enter a name for the pen : grass
- Color : green

```
to do-plots  
  set-current-plot "Totals"  
  set-current-plot-pen "turtles"  
  plot count turtles  
  set-current-plot-pen "grass"  
  plot count patches with [pcolor = green]  
end
```

Insérer *do-plots* dans *setup* et *go* (à la fin)